The Performance Evaluation Research Center (PERC)

Participating Institutions:

Argonne Natl. Lab.
Lawrence Berkeley Natl. Lab.
Lawrence Livermore Natl. Lab.
Oak Ridge Natl. Lab.
Univ. of California, San Diego
Univ. of Illinois
Univ. of Maryland
Univ. of Tennessee, Knoxville

Website: http://perc.nersc.gov

PERC Overview

- An “Integrated Software Infrastructure Center” (ISIC) sponsored under DoE’s SciDAC program.
- Funding: approx. $2.4 million per year.
- Mission:
  - Develop a science of performance.
  - Engineer tools for performance analysis and optimization.
- Focus:
  - Large, grand-challenge calculations, especially large-scale scientific codes used in SciDAC projects.

Specific Objectives

- Understand key factors in scientific codes that affect performance.
- Understand key factors in computer systems that affect performance.
- Develop models that accurately predict performance of codes on systems.
- Develop an enabling infrastructure of tools for performance monitoring, modeling and optimization.
- Validate these ideas and infrastructure via close collaboration with DOE Office of Science and others.
- Transfer the technology to end users.

Economic Benefits

Consider the economic value of improving the performance of a single high-end scientific application code by 20%.

Assume:

- $10 million computer system lease cost per year.
- $10 million per year in site costs, support staff, etc.
- 10-year lifetime of code.
- Code uses 5% of system cycles each year.

Savings: $2,000,000.

Scientific benefit (additional computer runs and research) is probably much higher.

Quantitative Feedback to Vendors

- We are invited by vendors to provide guidance on the design of current and future systems.

BUT

- At present we can provide only vague information – little if any quantitative data or rigorous analysis.

The performance monitoring and modeling capability being developed in PERC will significantly improve our ability to influence future scientific computer systems.

Four Thrusts

Better Benchmarks:

- Kernel benchmarks extracted from real codes reduce complexity of analyzing full-size benchmarks.
- Low-level benchmarks measure key rates of data access at various levels of memory hierarchy.

Modern performance monitoring tools:

- Flexible instrumentation systems capture hardware and software interactions, instruction execution frequencies, memory reference behavior, and execution overheads.
- An advanced data management infrastructure tracks performance experiments and data across time and space.

Four Thrusts, cont.

Performance modeling:

- *Application signature* tools characterize applications independent of the machine where they execute.
- *Machine signature* tools characterize computer systems, independent of the applications.
- *Convolution* tools combine application and machine signatures to provide accurate performance models.
- *Statistical models* find approximate performance models based on easily measured performance data.
- *Performance bound* tools determine ultimate potential of an application on a given system.

Four Thrusts, cont.

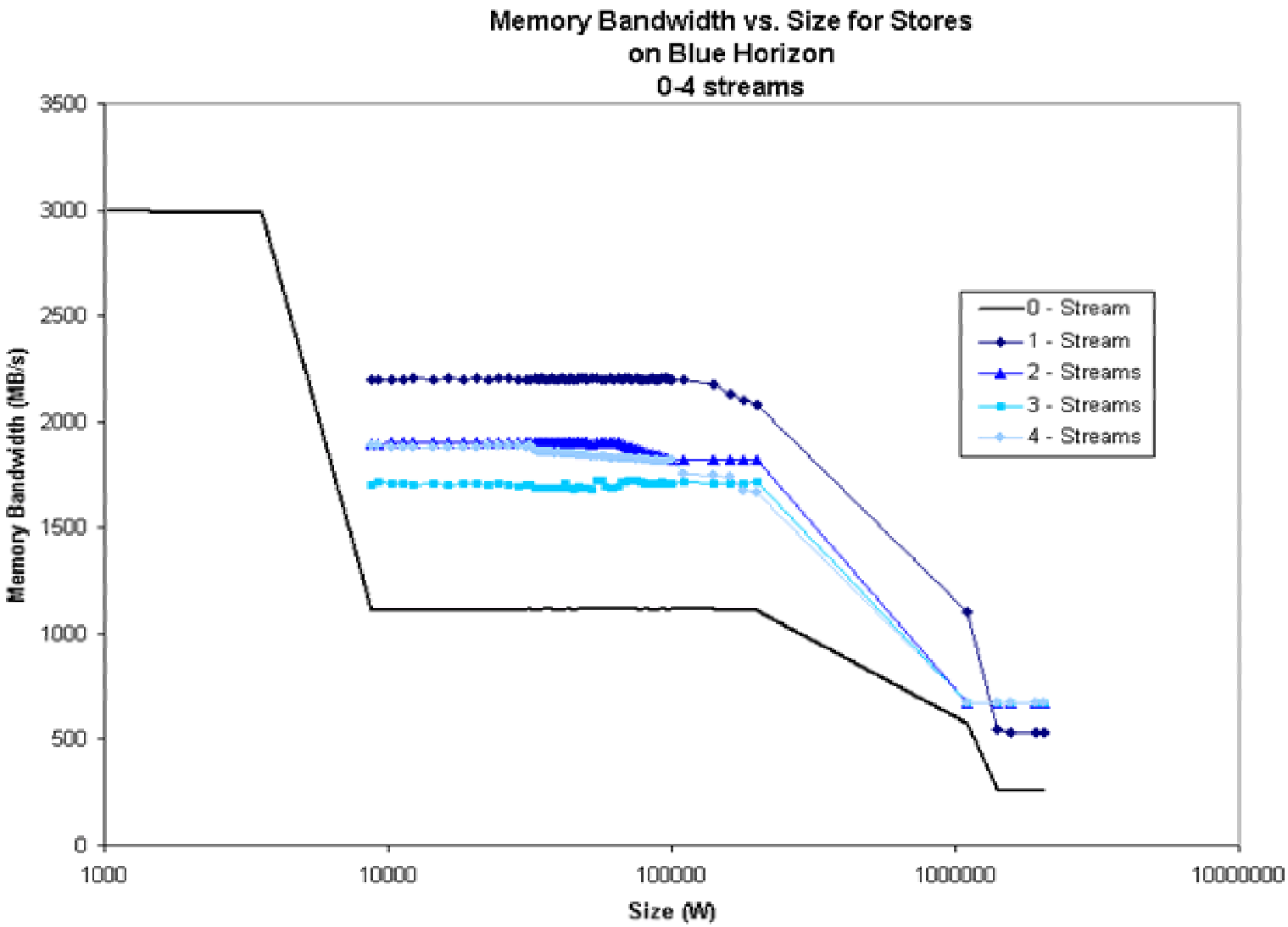
Performance optimization:

- *Compile-time optimization* mechanisms analyze source code to improve performance.
- *Self-tuning software* automatically tunes code based on real-time measurements of hardware environment.
- *Performance assertions* permit user-specified run-time tests to possibly change the course of the computation depending on results.
- *Performance portability* programming techniques to insure that code runs at near-optimal performance across a variety of modern systems.

Partnerships with SciDAC Scientific Projects

- Terascale Simulation of Neutrino-Driven Supernovae
- Advanced Computing for 21st Century Accelerators
- National Computational Infrastructure for Lattice Gauge Theory
- Collaborative Design and Development of Community Climate System Model for Terascale Computers
- Numerical Computation of Wave-Plasma Interactions
- The Plasma Microturbulence Project
- Terascale Optimal PDE Solvers

Memory Access Patterns (MAPS)

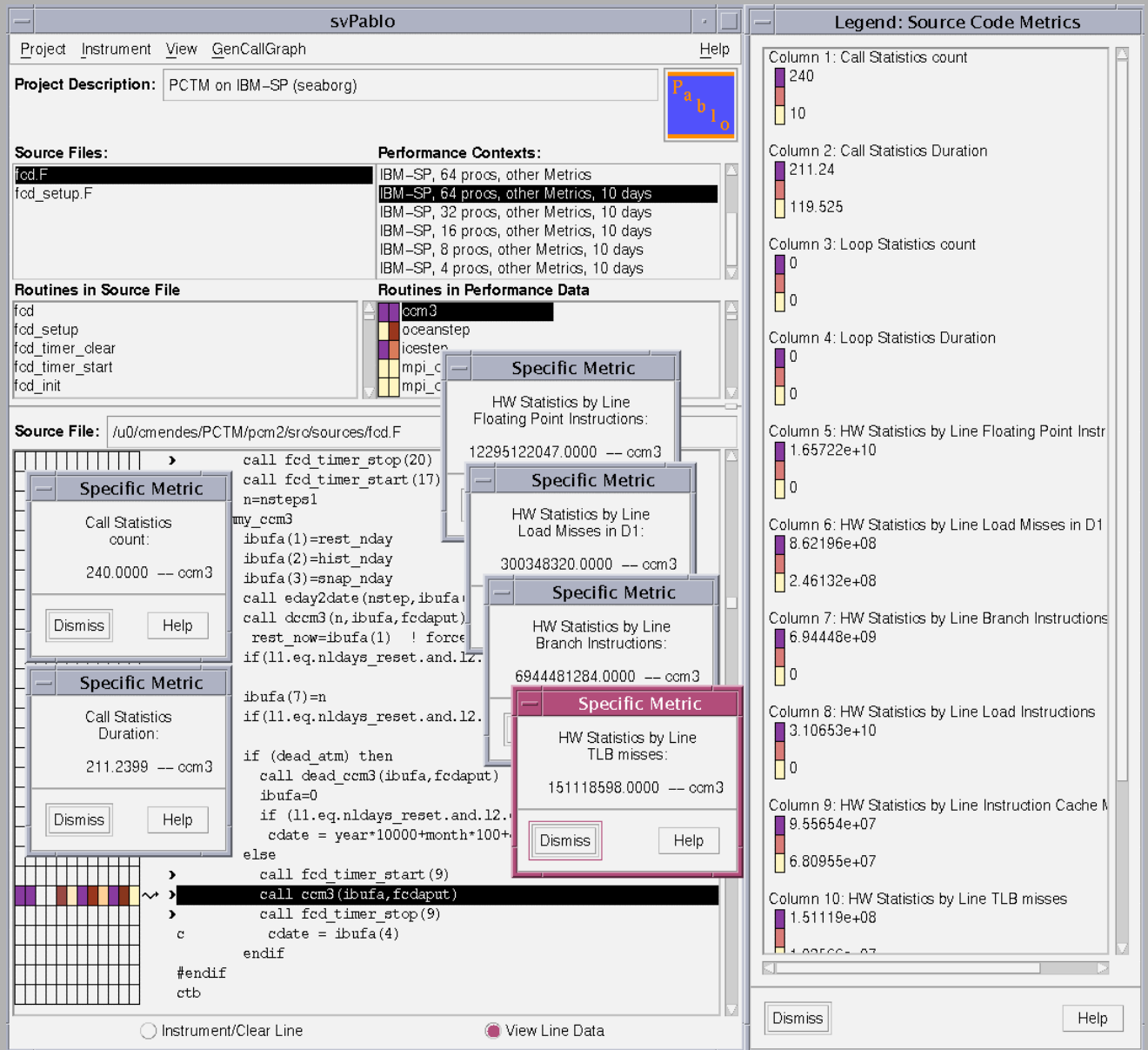


Performance Model Vs Actual

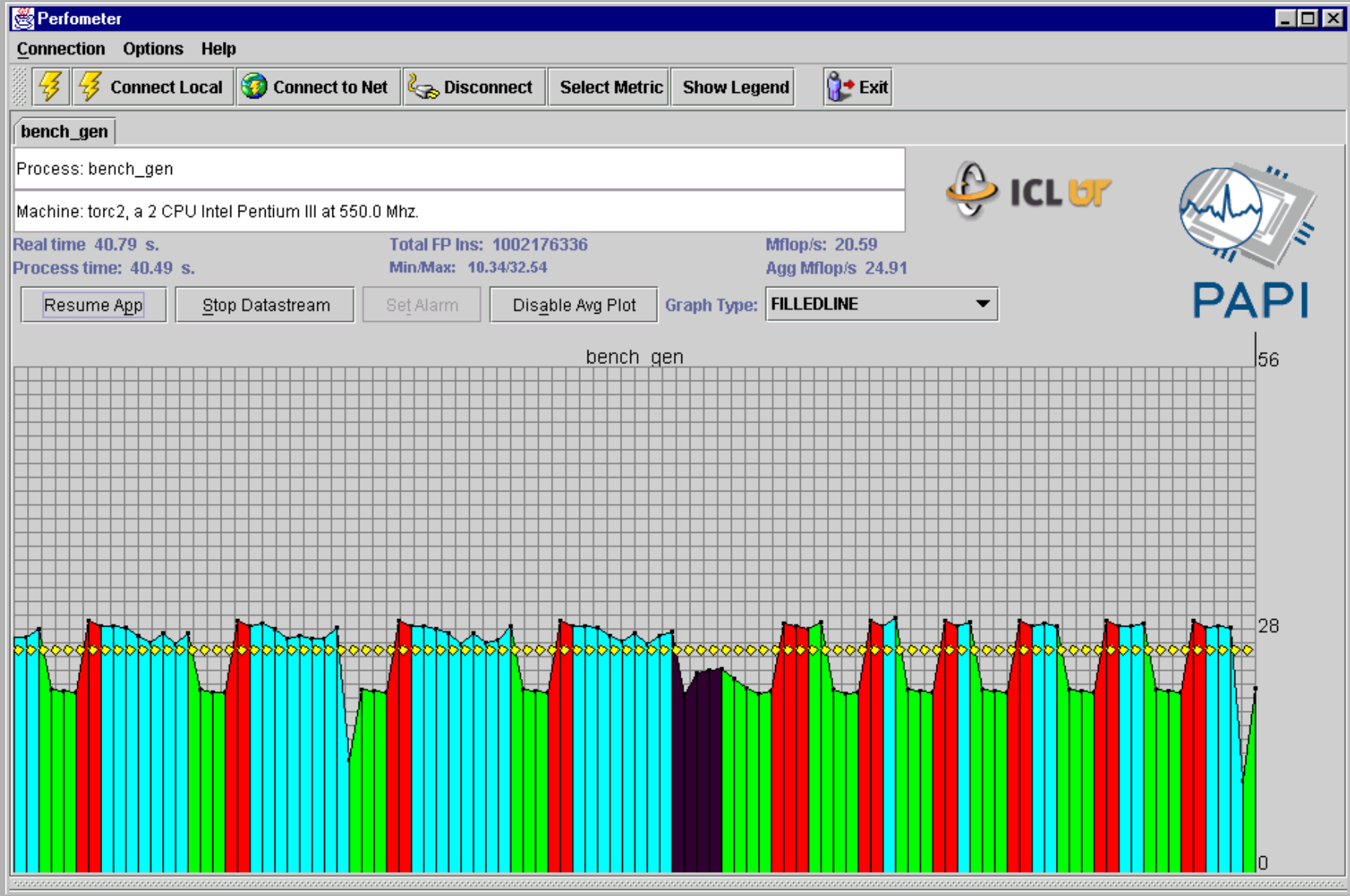
# CPUs	Real Time	Predicted Time	% Error
2	31.78	31.82	0.13
4	29.07	31.27	7.57
8	36.13	33.72	6.67
64	44.91	43.91	2.23
96	48.87	47.15	3.52
128	52.88	52.46	0.79

PETSc kernel, run on IBM SP at SDSC.

SvPablo Graphical Interface



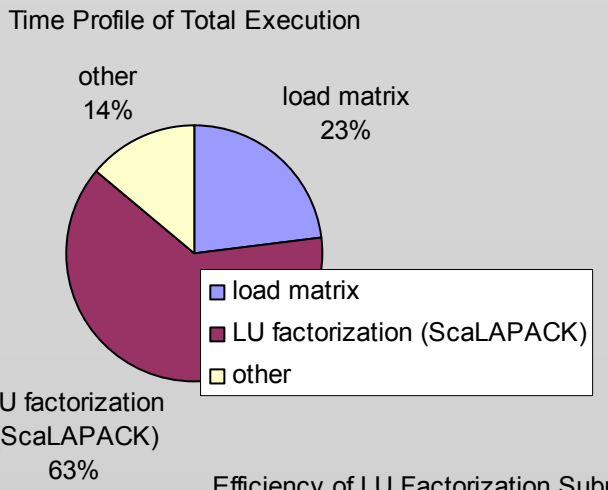
PAPI Perfometer Interface

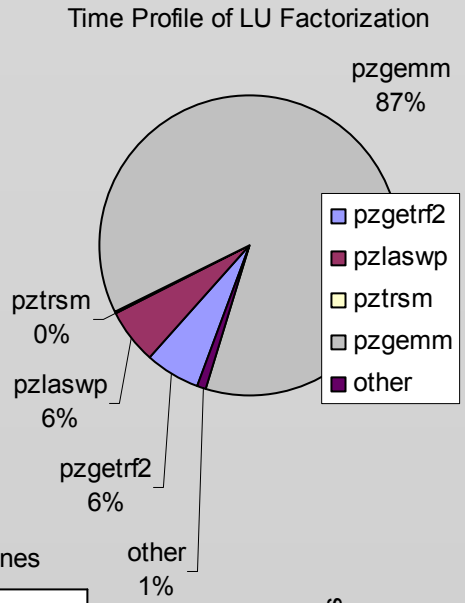


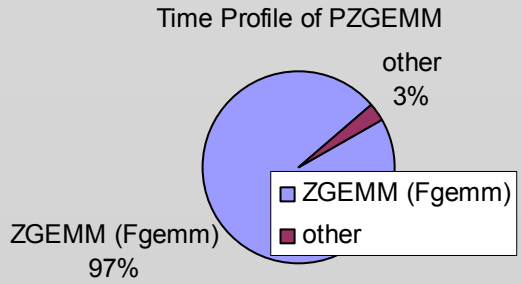
Performance Analysis

AORSA3D (fusion)

AORSA3D was ported and benchmarked on IBM and Compaq platforms. A detailed performance analysis has begun using SvPablo and PAPI. The results below are for a 400 Fourier mode run on 16 processors and 1 node of an IBM SP (Nighthawk II / 375MHz).

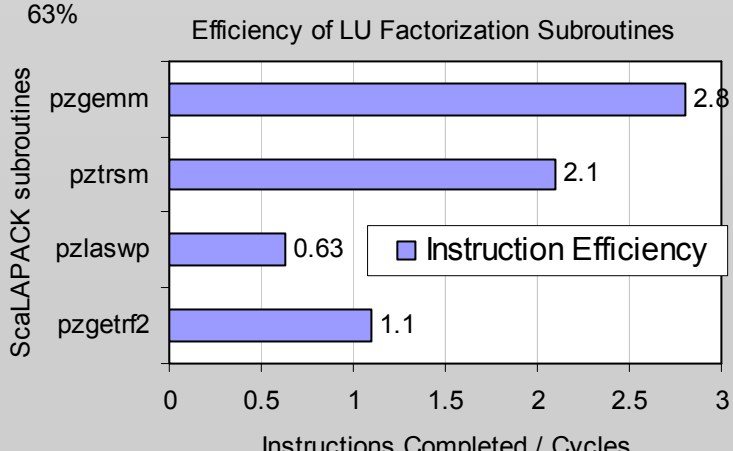


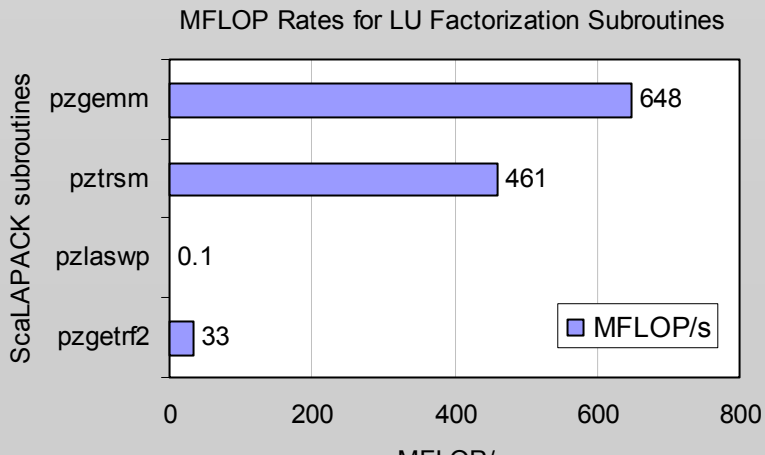




Performance for ZGemm

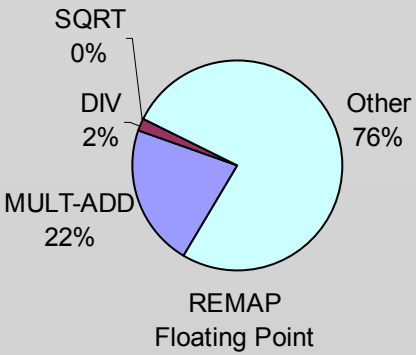
Denisty of Mem Access	1
Denisty of FLOPs	1.8
MFLOP/s	664
L1 cache hit rate	0.98
L2 cache hit rate	0.96
TLB misses	285034

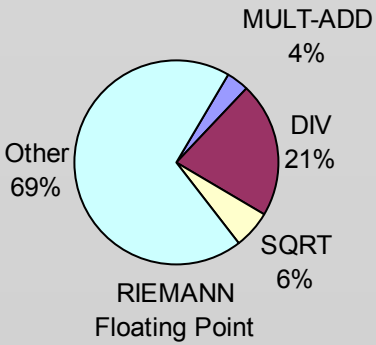


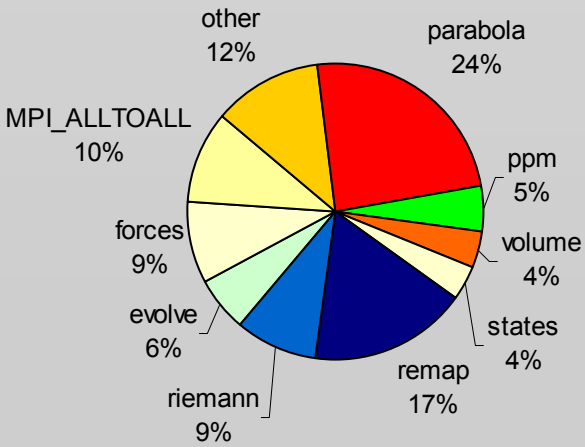


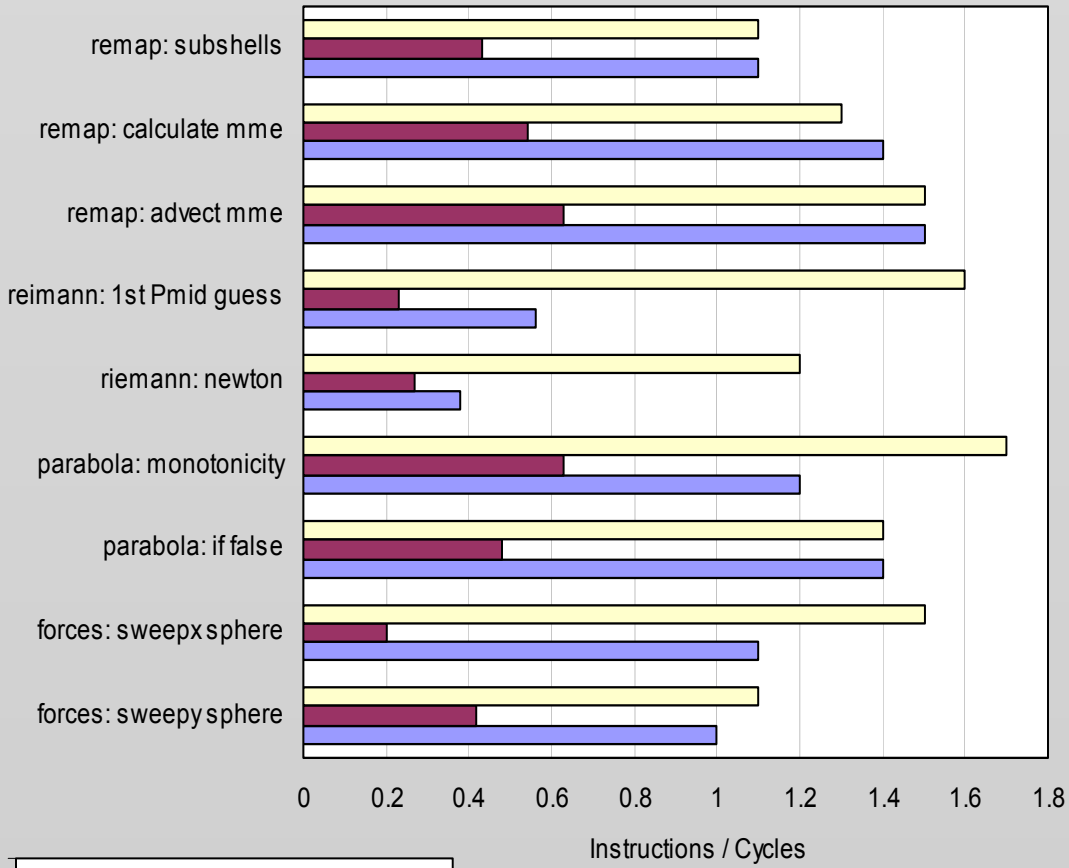
Performance Analysis
EVH1 (high-energy physics)

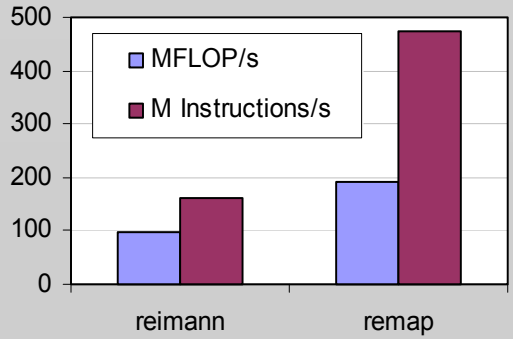
Aggregate performance measures over all tasks for a .1 simulation-second run. Collected with PAPI on an IBM SP (Nighthawk II / 375MHz).

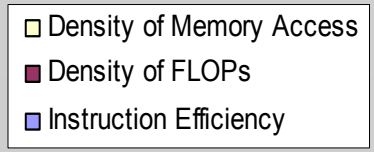








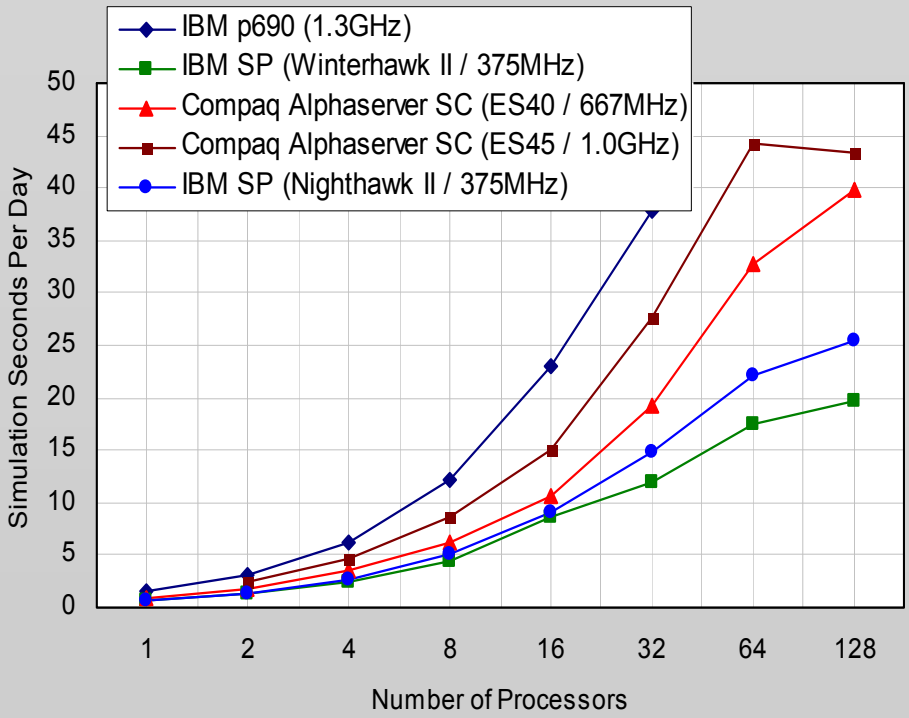


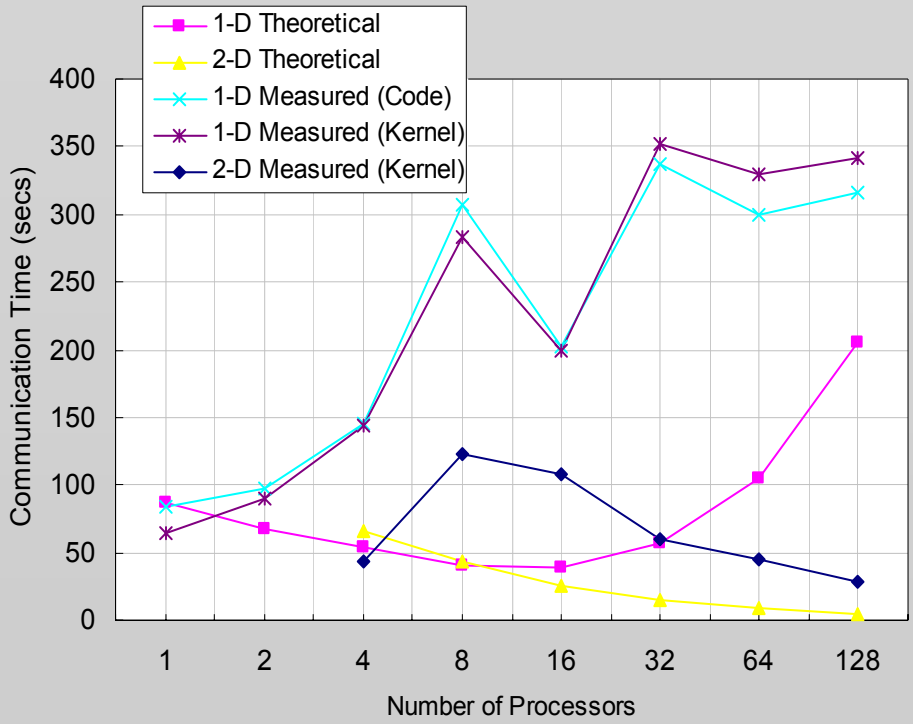


Analysis and Tuning
EVH1 (high-energy physics)

It was found that for more than 64 processors, EVH1 is communication-bound. The predominant routine (>50% of execution time) at this scale is MPI_ALLTOALL, which is used in matrix-transpose-like operations.

The current implementation uses a 1D decomposition for the matrix-array; a modeling and analysis study has shown that a 2D decomposition would result in a large improvement. Benchmarking and analysis results are below.

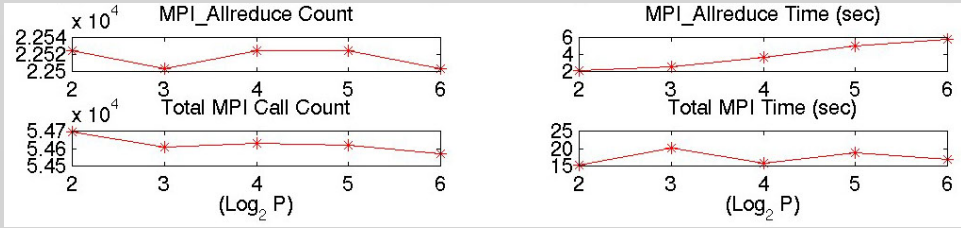


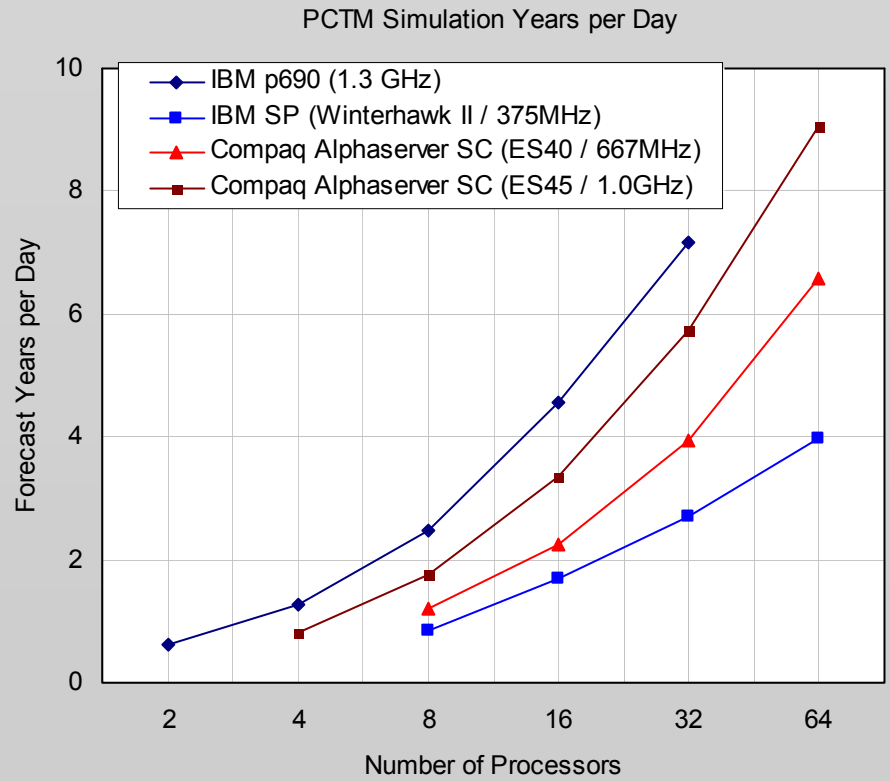


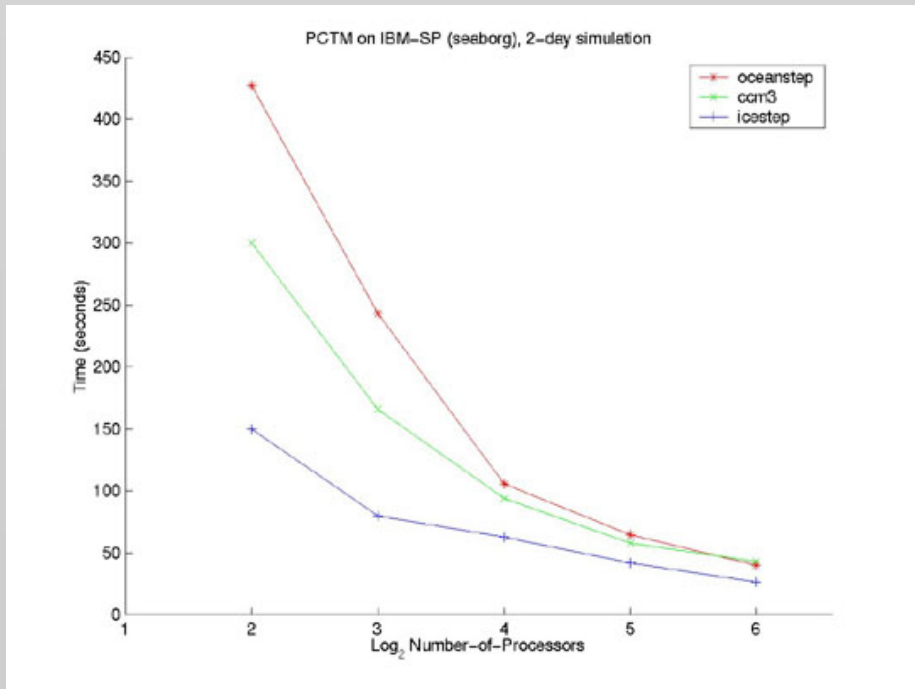
Performance Analysis

PCTM (climate modeling)

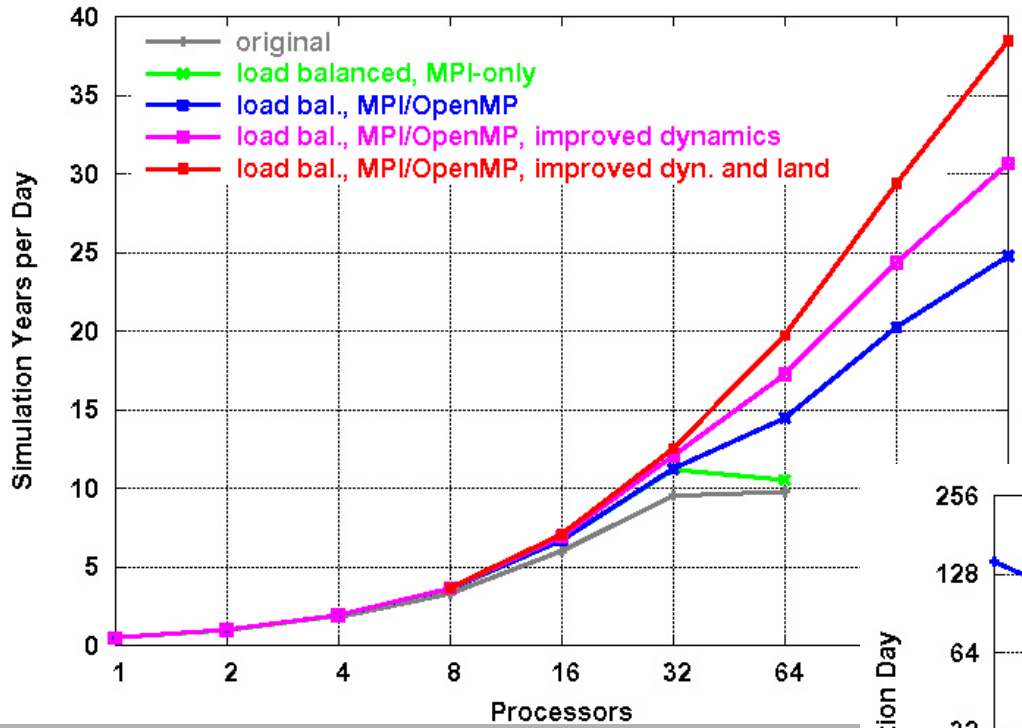
In studies with PCTM, we examined the impact of different domain decompositions, as well as using fewer MPI processes per SMP node. These studies indicated a strong performance dependence on message-passing performance, where using 64 MPI processes on 32 4-way SMP nodes (leaving 64 processors idle) was 30% faster than when running on 16 SMP nodes.





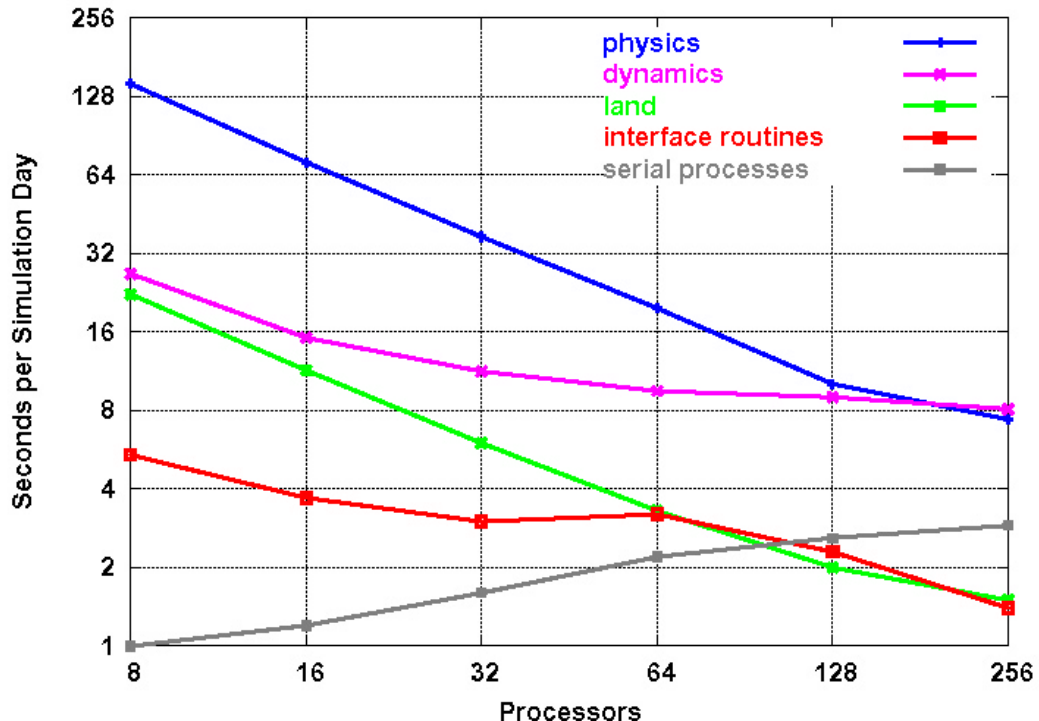


Analysis and Tuning
CAM (atmospheric modeling)



CAM performance measurements on IBM p690 cluster (and other platforms) were used to direct development process. Graph shows performance improvement from performance tuning and recent code modifications.

Profile of current version of CAM indicates that improving the serial performance of the physics is the most important optimization for small numbers of processors, and introducing a 2D decomposition of the dynamics (to improve scalability) is the most important optimization for large numbers of processors.



Working with PERC

- Benchmarking
  - Application group works with PERC to specify relevant benchmark codes and problems.
  - PERC characterizes performance, generates performance models, and suggests optimizations.
- Performance Tools
  - PERC trains application developers to use tools.
  - Application group uses tools in development, providing feedback on functionality and future development

For further information: http://perc.nersc.gov